Distributed sparse BSS for large-scale datasets

Tobias I Liaudat, Jerome Bobin, Christophe Kervazo

Distributed sparse BSS for large-scale datasets

Tobias I. Liaudat, Jérôme Bobin, and Christophe Kervazo
CEA, IRFU, SEDI/Service d'Astrophysique
91191 Gif-sur-Yvette Cedex, France
Email: tobiasliaudat@gmail.com

Abstract—**Blind Source Separation (BSS) [1] is widely used to analyze multichannel data stemming from origins as wide as astrophysics to medicine. However, existent methods do not efficiently handle very large datasets. In this work, we propose a new method coined DGMCA (Distributed Generalized Morphological Component Analysis) in which the original BSS problem is decomposed into subproblems that can be tackled in parallel, alleviating the large-scale issue. We propose to use the RCM (Riemannian Center of Mass – [6][7]) to *aggregate* during the iterative process the estimations yielded by the different subproblems. The approach is made robust both by a clever choice of the weights of the RCM and the adaptation of the heuristic parameter choice proposed in [4] to the parallel framework. The results obtained show that the proposed approach is able to handle large-scale problems with a linear acceleration performing at the same level as GMCA and maintaining an automatic choice of parameters.**

I. Large-scale Blind Source Separation

Given m row observations of size t stacked in a matrix $\mathbf{Y}$ assumed to follow a linear model $\mathbf{Y} = \mathbf{AS} + \mathbf{N}$, the objective of BSS [1] is to estimate the matrices $\mathbf{A}$ (size $m \times n$) and $\mathbf{S}$ (size $n \times t$) up to a mere permutation and scaling indeterminacy. In this model, $\mathbf{A}$ mixes the n row sources in $\mathbf{S}$, the observations being entached by some unknwown noise $\mathbf{N}$ (size $m \times t$). We will assume that $n \leq m$. While ill-posed, this problem can be regularized assuming the sparsity of $\mathbf{S}$ [2]. The estimation will then turn into the minization of:

$$\hat{\mathbf{A}}, \hat{\mathbf{S}} = \underset{\mathbf{A},\mathbf{S}}{\arg\min} \;\; \frac{1}{2} \left\|\mathbf{Y} - \mathbf{AS}\right\|_F^2 + \left\|\Lambda \odot \mathbf{S}\right\|_1 + i_{\mathbf{X}:\|\mathbf{X}_k\|_2=1,\,\forall k}(\mathbf{A}) \,, \tag{1}$$

with $\|\cdot\|_F$ the Frobenius norm, Λ the regularization parameters and $i_{\mathcal{C}}(\cdot)$ the indicator function of the set $\mathcal{C}$. The first term is a data fidelity one, the second enforces the sparsity and the last avoids degenerated solutions with $\|\mathbf{A}\|_F^2 \to 0$ by enforcing unit columns.

To tackle Eq. (1), the GMCA [4] algorithm has known a tremendous success due to an automatic decreasing parameter strategy making it robust. However, in this work we will assume that the data $\mathbf{Y}$ are *large-scale* in the sense that t can have huge values (e.g. up to 10^9 samples), which make the treatement of $\mathbf{Y}$ as a whole intractable. In this context, using GMCA or most other algorithms is prohibitive.

II. Proposed method

This difficulty motivates the construction of J subproblems (j) of the type $\mathbf{Y}_j = \mathbf{AS}_j + \mathbf{N}_j$ where j denotes a subset of t_j columns of the corresponding matrices. We use disjoints sets with $\sum_j |t_j| = t$. A natural idea is then the extension of GMCA to work in parallel on the tractable smaller subproblems to minimize Eq. (1). While this approach is reminiscent to mini-batch approaches in machine learning [9], it however raises two issues in the context solving BSS through GMCA: i) each subproblem (j) yields a full estimate $\hat{\mathbf{A}}_{(j)}$ of $\mathbf{A}$. Is it possible to *aggregate* them to get a better final estimate?; ii) is it possible to *extend the automatic parameter choice* of GMCA (that made its success) to a parallel implementation?

A naive approach would be to independantly solve each subproblem (j) and aggregate the different final results. However, since GMCA is an iterative algorithm, aggregating the estimations $\hat{\mathbf{A}}_{(j)}$ of the different subproblems (j) *during the iterations* should reduce the error propagation, thus improving the results over the naive approach. More specifically, our DGMCA algorithm performs the agreggation through the weighted RCM [6] of the different columns $\hat{\mathbf{a}}_{i,\,(j)}^{(k+1)}$ of the estimations $\hat{\mathbf{A}}_{(j)}^{(k+1)}$ yielded by the different (j) subproblems at iteration $k+1$, which enables to take into account the geometry of the problem and the fact that each column must respect the unit norm constraint. Its calculation is done following a gradient descend where its convergence is assured by [7]. Roughly speaking, the RCM can be understood as a weighted angular mean on the hypersphere. To robustify this process, we further propose to *compute the weights based on an estimation of the Signal-to-Noise Ratio (SNR)* of the corresponding estimated sources $\tilde{\mathbf{s}}_j^{i,\,(k+1)}$ to penalize noisy estimations (cf. Algorithm 1).

```
procedure DGMCA($\mathbf{Y}$, parameters)
    while do not converge do
        Calculate $\lambda_i^{(k)}$
        for $j = 1, \ldots, J$ do
            $\tilde{\mathbf{S}}_j^{(k+1)} \leftarrow \left(\hat{\mathbf{A}}_{RCM}^{(k)}\right)^{\dagger} \mathbf{Y}_j$    (LS estimation)
            $\hat{\mathbf{S}}_j^{(k+1)} \leftarrow \mathcal{S}_{\Lambda^{(k)}}\left(\tilde{\mathbf{S}}_j^{(k+1)}\right)$    (Prox. op. of $\|\Lambda^{(k)} \odot \cdot\|_1$)
            $\hat{\mathbf{A}}_{(j)}^{(k+1)} \leftarrow \mathbf{Y}_j \left(\hat{\mathbf{S}}_j^{(k+1)}\right)^{\dagger}$    (LS estimation)
            $\hat{\mathbf{a}}_{i,\,(j)}^{(k+1)} \leftarrow \frac{\hat{\mathbf{a}}_{i,\,(j)}^{(k+1)}}{\left\|\hat{\mathbf{a}}_{i,\,(j)}^{(k+1)}\right\|_2}, \forall i \in \{1, \ldots, n\}$    (Prox. op. of $i_{\mathcal{C}}(\cdot)$)
            $\left[\mathbf{W}_{RCM}^{(k+1)}\right]_{i,j} = \left\|\tilde{\mathbf{s}}_j^{i,\,(k+1)}\right\|_2^2 / \boldsymbol{\sigma}_{Y_j^i} \left\|(\hat{\mathbf{A}}^{(k)})^{\dagger}\right\|_F^2, \forall i, j.$
        Correct permutations in $\mathbf{A}_{(j)}^{(k+1)}$, $\forall j$.
        $\hat{\mathbf{A}}_{RCM}^{(k+1)} \leftarrow$ RCM($\hat{\mathbf{A}}_{(1)}^{(k+1)}, \ldots, \hat{\mathbf{A}}_{(J)}^{(k+1)}, \mathbf{W}_{RCM}^{(k+1)}$)    (Aggregation)
        $k \leftarrow k + 1$
    return $\hat{\mathbf{A}}_{RCM}^{(k)}$, $\hat{\mathbf{S}}^{(k)}$
```

Fig. 1. DGMCA. the operator $(\cdot)^{\dagger}$ is the pseudo-inverse, $\mathcal{S}_{\lambda}(\cdot)$ is the soft-thresholding operator with the threshold λ, $\mathbf{a}_i$ denotes the i column of $\mathbf{A}$ and the subscript (j) denotes the estimation of the j subproblem.

Concerning question ii), the parameter choice of GMCA needs to access the whole distribution of the sources at each iteration, which is intractable in the large-scale regime. We propose a new strategy using a parametrized exponential decay which adapts to the signal statistics by using the maximum value of the estimated sources also making it parallelizable. The threshold decay is regulated by the parameter α_i and can be adjusted *in the first iterations* by fitting a generalized Gaussian to the sources.

III. Empirical results and conclusions

Numerical experiments can be found in Fig.2 and 3. In brief, our method paves the way for *distributed* approaches of BSS problems with *automatic parameter tunning*. It not only allows to handles *large datasets* but it enables a *linear acceleration*. Furthermore, it does not lower the separation quality compared to GMCA outperforming methods like the optimized ODL [9].

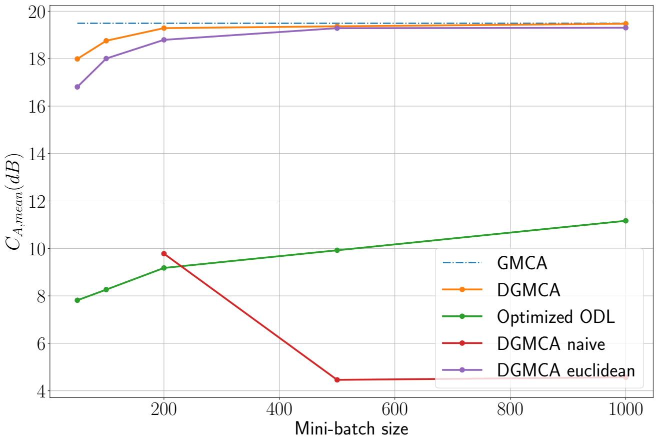

Fig. 2. Performance comparison of the GMCA, the ODL (Online Dictionary Learning [9]), and the DGMCA and its variants. Note that the parameters of the ODL algorithm have been optimized for this experiment by an exhaustive search. The x-axis corresponds to the size t_j of each subproblem (j), which is set for all j to be $t_j = t/J$. The y-axis represents the separation quality, measured by a mixing matrix criterion [5] defined as $-10\log\|\mathbf{P}\hat{\mathbf{A}}^{\dagger}\mathbf{A}^{\star} - \mathbf{I}\|_{\ell_1}$, where $\mathbf{A}^{\star}$ is the ground truth and $\mathbf{P}$ accounts for the correction of the permutations. To generate the experiments, the source matrix was randomly sampled from a Generalized Gaussian distribution with several profile parameters β between 0.35 and 1.4, having $n = 10$ sources, $t = 10000$ samples and $m = 20$ observations. The noise matrix $\mathbf{N}$ was set to have a SNR of 15dB. The matrix $\mathbf{A}$ is random and with a condition number fixed to 10. The experiment is repeated 3 times and the mean of the results is being plotted.

Four parallelized algorithms are compared: the presented *DGMCA*, the *DGMCA naive* method consisting of solving the J subproblems independently until convergence and performing the aggregation at the end, the *DGMCA Euclidean* method where the RCM is substituted with an Euclidean mean for the aggregation, and the ODL with its hyperparameters optimized. The results of the different algorithms are benchmarked with the GMCA using the entire observation matrix (which is only possible due to the relatively small t that we chose for the sake of the comparison).

The DGMCA outperforms the other parallelized methods maintaining a similar performance compared to the GMCA. It is worth to remark that the performance is limited by the size of the mini-batch and not by the total size t which can be increased thus making the number of mini-batches increase. The separation quality is indeed only reduced for extremely small t_j, which was expected due to the lack of statistics for the algorithm to work. The *DGMCA naive* is not plotted for the two smallest t_j as some mini-batches only contained noise, which caused the algorithm not to converge for the threshold level used (as the subproblems are solved independently until convergence). In addition, the huge gap between DGMCA and its naive version confirms the usefulness of using an aggregation process during the iterations. Furthermore, using the RCM as aggregation and therefore taking into account the geometry of the problem enables better results than with an Euclidian mean. In the context of reproducible research, the code is available online at: https://github.com/tobias-liaudat/DGMCA.

Acknowledgements

This work is supported by the European Community through the grant LENA (ERC StG - contract no. 678282).

References

[1] P. Comon and C. Jutten, "Handbook of Blind Source Separation: Independent component analysis and applications," Academic press, 2010.

[2] M. Zibulevsky and B. A. Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary," *Neural computation*, vol. 13, no. 4, pp. 863882, 2001.

[3] A. Cichocki and R. Zdunek, "Regularized alternating least squares algorithms for non-negative matrix/tensor factorization," in *International Symposium on Neural Networks*. Springer, 2007, pp. 793802.

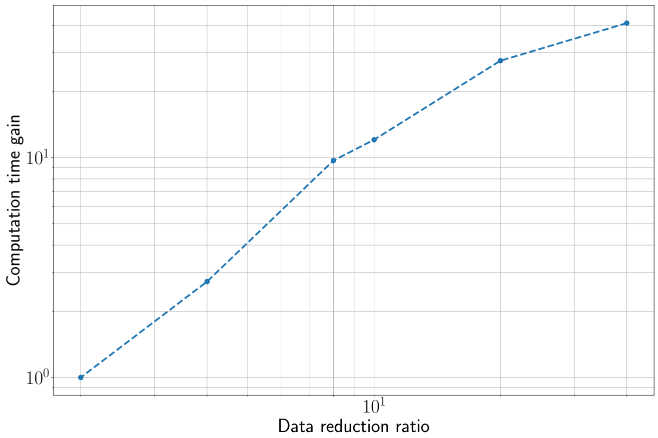

Fig. 3. Computational time gain between the parallelized DGMCA and the GMCA algorithms against the data reduction ratio which is calculated as the problem total size, t, divided by the size of the mini-batch, $t/t_j = J$. Each point on the figure represents the mean over 10 problems. The experiment was run using a C++ parallelized version of the DGMCA algorithm and the maximum number of mini-batches used is 40 as it is the number of cores the computer cluster used had. The setup of the experience is similar to the one in Fig 1, with a β parameter of 0.5, having $n = 5$ sources, $t = 10000$ samples and a SNR of 40dB. The linear trend of the time gain was predicted by the complexity analysis of the algorithms, and now confirmed by the numerical experiment.

[4] J. Bobin, J.-L. Starck, J. Fadili, and Y. Moudden, "Sparsity and morphological diversity in blind source separation," *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2662–2674, 2007.

[5] J. Bobin, J. Rapin, A. Larue, and J.-L. Starck, "Sparsity and adaptivity for the blind separation of partially correlated sources," *IEEE Transactions on Signal Processing*, vol. 63, no. 5, pp. 1199–1213, 2015.

[6] B. Afsari, "Riemannian Lp center of mass: Existence, uniqueness, and convexity," Proc. Amer. Math. Soc., 139:655673, 2011.

[7] B. Asfari, R. Tron and R. Vidal, "On The Convergence of Gradient Descent for Finding the Riemannian Center of Mass," arXiv:1201.0925 (Dec 2011)

[8] J. Munkres, "Algorithms for the Assignment and Transportation Problems," Journal of the Society of Industrial and Applied Mathematics, 5(1):32-38, March 1957.

[9] J. Mairal, F. Bach, J. Ponce and G. Sapiro, "Online Learning for Matrix Factorization and Sparse Coding," *Journal of Machine Learning Research*, 2010, 11 (1), pp.1960.